



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/887,576	06/25/2001	Irit Loy	LOY=3	7562

1444 7590 06/09/2004  
BROWDY AND NEIMARK, P.L.L.C.  
624 NINTH STREET, NW  
SUITE 300  
WASHINGTON, DC 20001-5303

EXAMINER

LU, KUEN S

ART UNIT	PAPER NUMBER
----------	--------------

2177

DATE MAILED: 06/09/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**  
*Supplemental*

Application No.

09/887,576

Applicant(s)

LOY ET AL.

Examiner

Kuen S Lu

Art Unit

2177

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 19 March 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-69 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-69 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date 4,5,7,8/10-1-01.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

***This is supplemental non-final***

***DETAILED ACTION***

***Response to Amendment***

1. The Examiner noted the Applicants' amendment "AMENDMENT AND REMARKS", filed on March 19, 2004. As per Claim Rejections under U.S.C. 35 §103(a), please refer to "***Claim Rejections - 35 USC § 103***" as stated below. As per Applicant's REMARKS, please refer to the section "***Response to Arguments***" after the section "***Claim Rejections - 35 USC § 103***".

***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 1-9, 17-21, 23-31, 39-43, 45-53 and 61-69 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunC22 (Sun Cluster 2.2 Software Installation Guide, Sun Microsystems, Inc. Part Number 806-1008, April 1, 1999, hereafter "SunC22") and in view of Cramer et al. (U.S. Patent 5,946,685).

As per claim 1, 23 and 45, SunC22 teaches "a cluster of computing nodes having shared access to one or more file system in data storage using parallel file system software" at Pages C-5 and 24 where global file systems are mounted on the logical host system. The file system is globally mounted and shared by each node of the cluster. The file system is thus a parallel file system.

SunC22 teaches "initiating a session of a data management application on a first one of the nodes" at Page 5-14 for bring up an Oracle database instance into service.

SunC22 teaches parallel file system as previously described.

SunC22 does not specifically teach receiving a request submitted to the file system software at a second one of the nodes to mount one of the file systems in the data storage on the second one of the nodes.

However, Cramer teaches receiving a request submitted to the file system software at a second one of the nodes to mount one of the file systems in the data storage on the second one of the nodes at col. 9, lines 47-53 by either determining the server for the mount point or requesting the server to generate the mount point and link to the server-side vnode corresponding to the mount point.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Cramer's teaching with SunC22's because both references are devoted to mount file system to be shared by a plurality of computer nodes either distributed or in parallel (Cramer: col. 3, lines 34-35, SunC22: Page C-5). The combined reference would have provided more flexibility on mounting or unmounting global file systems of cluster system where file systems are mainly

mounted during the cluster system startup (Page C-5). Furthermore, the unmount of file system requires some additional steps to stop the processes currently running on the file systems (SunC22: Page C-5).

Cramer further teaches "sending a mount event message from the second node to the first node responsive to the request, for processing by the data management application on the first node" at col. 9, lines 59-60 by returning the information concerning the mount point to the requesting node.

As per claims 2, 24 and 46, Cramer further teaches the following:

"mounting first and second instances of the one of the file systems on the first and second nodes, respectively, responsive to the mount event message" at col. 9, lines 47-53 by requesting the server to generate the mount point and link to the server-side vnode corresponding to the mount point on the first request to mount an unmounted file system or determining the server for the mount point on the second or additional requests for mounting the already mounted file system.

As per claims 3, 25 and 47, Cramer further teaches "receiving a further request at the second node to unmount the second instance of the one of the file systems at the second node" at col. 12, lines 3-4 by a process associated with a node receiving a global unmount command;

"sending, responsive to the further request, a pre-un-mount event message to the first node" at col. 12, lines 22-31 by the list server of the mount point for having each node un-splice the file system resource from the node's mount point proxy vnode; and

“responding to the pre-un-mount event message so as to permit unmounting of the second file system instance without unmounting the first file system instance” at col. 12, lines 29-30 and 32-33 by the list server of the mount point to delete the VFS from the global mount list and releasing the global lock representing the mount point.

As per claims 4, 26 and 48, Cramer further teaches “responding to the pre-un-mount event message comprises determining at the first node, responsive to one or more flags set in the pre-un-mount event message, whether the request was submitted on the first node or on another one of the nodes” at col. 12, lines 22-31 by the list server of the mount point for having each node un-splice the file system resource from the node’s mount point proxy vnode and informing other nodes to delete their VFS and PxFobj data structure, thus the originating node is spare of the information and determines the original node of the unmount command.

As per claims 5, 27 and 49, Cramer further teaches “receiving the pre-un-mount event message at the first node” at col. 12, lines 22-31 by the list server of the mount point for having each node un-splice the file system resource from the node’s mount point proxy vnode implies the first node receives the un-splice message;

“obtaining a data management access right from a physical file system (PFS) software component at the first node responsive to the pre-un-mount event message” at col. 12, lines 23-26 by un-splicing the file system resource and deleting the contents of the mounted\_here VFS pointer; and

“processing the pre-un-mount event message using the access right” at col. 12, lines 27-31 by deleting the infrastructure used to support the unmounted resource, VFS

pointer of the global mount point and informing other nodes to delete their VFS and PxFobj data structure.

As per claims 6, 28 and 50, Cramer further teaches "receiving the request comprises receiving first and second requests to mount different ones of the file systems in the data storage" at col. 11, lines 5-7 by global mount procedure to call the list server with information on newly generated object and the resource used to instantiate it and the list server adds the object to the cluster-wide global mount list, "wherein receiving the further request comprises receiving further first and second requests to unmount the different ones of the file systems" at col. 12, lines 3-4 by a process associated with a node receiving a global unmount command which may come from different nodes for different file systems, and "the pre-un-mount event message comprises, responsive to dispositions set for the different ones of the file systems" at col. 12, lines 22-31 by the list server of the mount point for having each node un-splice the file system resource from the node's mount point proxy vnode for each unmount request from a node; and "sending a first pre-un-mount event message to the first node responsive to the first unmount request, and sending a second pre-un-mount event message 1 responsive to the second unmount request to a further node, on which a further data management application session has been initiated" at col. 12, lines 22-31 by the list server of the mount point for having each node un-splice the file system resource from the node's mount point proxy vnode and informing other nodes to delete their VFS and PxFobj data structure each time a list server node responding to the unmount request from a node.

As per claims 7, 29 and 51, Cramer further teaches "responding to the pre-un-mount event message comprises sending a reply to the message from the first node to the second node, and comprising, responsive to the reply, unmounting the second file system instance and sending an unmount event message from the second node to the first node" at col. 12, lines 22-31 by the list server having each node un-splicing file resource from that node's mount proxy vnode and the mount mechanism will **then** delete the infrastructure used to support the unmounted resource, thus the second node is synchronizing every node's, including the first node's, response of un-splicing file resource, and at col. 12, lines 32-33 by releasing the lock representing the mount point.

As per claims 8, 30 and 52, Cramer further teaches "determining at the first node, responsive to one or more flags set in the unmount event message, whether the further request was submitted on the first node or on another one of the nodes" at col. 5, lines 15-23 by pairing proxy vnodes on each client node with corresponding vnodes on the server side, the global unmount mechanism utilizes the vnode link to determine the node of the submitting request and at col. 12, lines 22-31 by the list server of the mount point for having each node un-splice the file system resource from the node's mount point proxy vnode and informing other nodes to delete their VFS and PxObj data structure, thus the originating node is spare of the information and determines the original node of the unmount command.

As per claims 9, 31 and 53, Cramer further teaches "determining at the first node, responsive to one or more flags set in the mount event message, whether the further request was submitted on the first node or on another one of the nodes" at col. 9, lines



47-53 by the list server generating a provider object for the requesting node and the object is returned to the requesting node for further generating an associated cache object, thus determines the originating node which initiates the mount command and at col. 5, lines 15-23 by pairing proxy vnodes on each client node with corresponding vnodes on the server side, the global mount mechanism utilizes the vnode link to determine the node of the submitting request.

As per claims 17, 39 and 61, Cramer further teaches "receiving a response to the mount event message from the data management application on the first node" at Fig. 6, elements 204, 206 and 208, col. 11, lines 22-27 by spicing the resource of mount point in each node; and "mounting an instance of the one of the file systems on the second node subject to the response from the data management application on the first node" at col. 11, lines 27-32 by linking covered\_vnode pointer of the proxy VFS to the vnode of the mount point, indicating a file system has been mounted **when** the mounted\_here pointer is set.

As per claims 18, 40 and 62, Cramer further teaches "receiving a further request submitted to the parallel file system software to mount the one of the file systems on a further one of the nodes" at col. 13, lines 15-19 by providing a mount point with global locking capability, and "sending a further mount event message responsive to the further request for processing by the data management application on the first node" at col. 13, lines 44-57 by giving write access to the global lock corresponding to the mount point.

As per claims 19, 41 and 63, Cramer further teaches "the further one of the nodes is the first node" at col. 13, lines 15-19 by mounting file system to any node making such request.

As per claims 20, 42 and 64, Cramer further teaches "receiving first and second unmount requests to unmount the file system from the second node and from the further one of the nodes" at col. 12, lines 3-10 by invoking the unmount command from any node, and "generating first and second pre-un-mount event messages at the second node and at the further one of the nodes responsive to the first and second unmount requests, for processing by the data management application on the first node" at col. 12, lines 22-31 by having each node un-splicing the file system resource from that node's proxy vnode mount point and then deleting the supporting infrastructure.

As per claims 21, 43 and 65, Cramer further teaches "sending a reply to the first and second pre-un-mount event messages from the first node to the second node and to the further one of the nodes, and, responsive to the reply, unmounting the file system at the second node and the further one of the nodes, and generating respective unmount event messages at the second node and at the further one of the nodes" at col. 12, lines 27-33 by deleting the mount point's supporting infrastructure and informing other nodes to delete their VFS and OxFobj structures.

As per claims 67, 68 and 69, Cramer further teaches "request to mount one of the file systems is submitted by a user application running on the second one of the nodes" at col. 9, lines 47-53 by either determining the server for the mount point or requesting the

server to generate the mount point and link to the server-side vnode corresponding to the mount point.

4. Claims 10-16, 32-38 and 54-60 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunC22 (Sun Cluster 2.2 Software Installation Guide, Sun Microsystems, Inc. Part Number 806-1008, April 1, 1999, hereafter "SunC22") in view of Cramer et al. (U.S. Patent 5,946,685), as applied to claims 1, 23 and 45 above, and further in view of Dugan et al. (U.S. Patent 6,363,411, hereafter "Dugan").

As per claims 10, 32 and 54, SunC22 does not specifically teach "initiating a session of a data management application".

SunC22 teaches "initiating a session of a data management application" at Page 5-14 for bring up an Oracle database instance into service.

SunC22 does not specifically teach "receiving a request and sending the mount event message".

However, Cramer teaches "receiving a request and sending the mount event message" at col. 9, lines 47-53 by either determining the server for the mount point or requesting the server to generate the mount point and link to the server-side vnode corresponding to the mount point and at col. 9, lines 59-60 by returning the information concerning the mount point to the requesting node.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Cramer's teaching with SunC22's because both references are devoted to mount file system to be shared by a plurality of computer nodes either distributed or in parallel (Cramer: col. 3, lines 34-35, SunC22: Page C-5).

The combined reference would have provided more flexibility on mounting or unmounting global file systems of cluster system where file systems are mainly mounted at the cluster startup (Page C-5). Furthermore, the unmount of file system requires some additional steps to stop the processes currently running on the file systems (SunC22: Page C-5).

The combined Cramer-SunC22 reference does not specifically teach **specifically using DMAPI** for initiating the session, receiving the request and sending the mount event message.

However, Dugan teaches using DMAPI for making a request for data as the DMAPI providing a common message set for all DM clients to interface with Data Management at Fig. 5(f), elements 410 and 412, and col. 48, lines 12-15.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Dugan's teaching with Cramer and SunC22's references by using standard set of DMAPI interface functions for session management because DMAPI would have enhanced centralizing and standardizing the data management which starts with creating a session on a node and specifying the Data Management events to be reported to the session.

As per claims 11, 33 and 55, Cramer further teaches "receiving an unmount request to unmount the file system from the second node" at col. 12, lines 3-4 by a process associated with a node receiving a global unmount command and "sending a pre-unmount event message to the first node" at col. 12, lines 22-31 by the list server of the

mount point for having each node un-splice the file system resource from the node's mount point proxy vnode.

The combined Cramer-SunC22 reference does not specifically teach **specifically using DMAPI** for receiving an unmount request to unmount the file system or sending a pre-un-mount event message to the first node.

However, Dugan teaches using DMAPI for providing a common, standard set of interface for receiving and sending data, and further utilizing local cache for data retrieval at col. 48, lines 12-21.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Dugan's teaching with Cramer and SunC22's references by using a common message set of DMAPI to interface with Data Management server such that receiving and sending message would be handled by a set of standardized functions which would simplify the programming effort for the applications.

As per claims 12, 34 and 56, Cramer further teaches "responding to the pre-un-mount event message comprises sending a reply to the message from the first node to the second node, and comprising, responsive to the reply, unmounting the second file system instance and sending an unmount event message from the second node to the first node" at col. 12, lines 22-31 by the list server having each node un-splicing file resource from that node's mount proxy vnode and the mount mechanism will **then** delete the infrastructure used to support the unmounted resource, thus the second node

is synchronizing every node's, including the first node's, response of un-splicing file resource, and at col. 12, lines 32-33 by releasing the lock representing the mount point.

The combined Cramer-SunC22 reference does not specifically teach **specifically using DMAPI** for sending a reply to the pre-un-mount event message, responsive to the reply, or sending an unmount event message.

However, Dugan teaches using DMAPI to process data management on a computer node at Fig. 5(f), elements 420 and 425, col. 48, lines 15-21.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Dugan's teaching with Cramer and SunC22's references such that the actual data application initiated by a client system could communicate with service administration system for sending or receiving data through a common message set of DMAPI, and further facilitating the data transfer by using cache storage for local data.

As per claims 13, 35 and 57, Dugan further teaches DMAPI interfacing the SA (System Administration) client and the data distribution process of SA, and a data management server which handles data extract at col. 48, lines 1-5.

As per claims 14, 36 and 58, Dugan further teaches using DMAPI for making a request for data as the DMAPI providing a common message set for all DM clients to interface with Data Management at Fig. 5(f), elements 410 and 412, and col. 48, lines 12-15.

As per claims 15, 37 and 59, Cramer further teaches sending event message at col. 12, lines 22-31 by the list server of the mount point for having each node un-splice the file system resource from the node's mount point proxy vnode.

The combined Cramer-SunC22 reference does not specifically teach **specifically using DMAPI** for sending the event message comprises setting one or more flags in the message to indicate whether the request was submitted on the first node or on another one of the nodes.

However, Dugan teaches using DMAPI for providing a common, standard set of interface for receiving and sending data, and further utilizing local cache for data retrieval at col. 48, lines 12-21.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Dugan's teaching with Cramer and SunC22's references by using a common message set of DMAPI to interface with Data Management server such that receiving and sending message would be handled by a set of standardized functions which would simplify the programming effort for the applications.

As per claims 16, 38 and 60, Cramer further teaches "sending a mount event message from the second node to the first node responsive to the request, for processing by the data management application on the first node" at col. 9, lines 59-60 by returning the information concerning the mount point to the requesting node.

The combined Cramer-SunC22 reference does not specifically teach invoking a DMAPI function "to obtain mount information regarding the one of the file systems, and

wherein in a response provided by the function, one or more flags are set to indicate whether the one of the file systems is mounted on the first node or on another one of the nodes in the cluster or on both the first node and on another one of the nodes in the cluster”.

However, Dugan teaches using DMAPI for providing a common, standard set of interface for receiving and sending data, and further utilizing local cache for data retrieval at col. 48, lines 12-21.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Dugan's teaching with Cramer and SunC22's references by using a common message set of DMAPI to interface with Data Management server such that receiving and sending message would be handled by a set of standardized functions which would simplify the programming effort for the applications.

5. Claims 22, 44 and 66 are rejected under 35 U.S.C. 103(a) as being unpatentable over SunC22 (Sun Cluster 2.2 Software Installation Guide, Sun Microsystems, Inc. Part Number 806-1008, April 1, 1999, hereafter “SunC22”) , in view of Cramer et al. (U.S. Patent 5,946,685), as applied to claims 1, 23 and 45 above, and further in view of Vahalia et al. (U.S. Patent 6,192,408, hereafter “Vahalia”).

As per claims 22, 44 and 66, the combined Cramer-SunC22 reference does not specifically teach “initiating the session of the data management application comprises initiating a data migration application, so as to free storage space on at least one of the volumes of data storage”, although Cramer teaches initiating a session of a data



management application at col. 9, lines 22-23 by a user associated with a process issuing a global mount command at a requesting node.

However, Vahalia teaches migrating file system from a failed data processor to a spare processor at Fig. 27, elements 241', 242', through 245', col. 28, lines 22-25 by migrating files owned by a failed processor to a spare data processor in a system that uses remote dual copy instead of a cached disk storage subsystem.

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine Vahalia's teaching with Cramer and SunC22's references by implementing the file system migration plan not just for a failed processor, but also for a routine data backup because such implementation would ensure data availability even in a reverse situation.

#### ***Response to the Arguments***

6. Applicant's arguments with respect to U.S.C. 102(e) rejections to claims 1-66 have been considered, for the Examiner's response, please see supplemental non-final Office Action as stated above.

a). At Pages 18-20, the Applicant mainly argued the Cramer reference teaches "**distributed file system**" as the file system and "**client/server model**" as the basic model to mount file system, further, upon considering above argument, especially the independent claims 1, 18 and 35, a new SunC22 reference was introduced, for its teaching on parallel file system mounted on cluster system (SunC22: Page 24, Paragraph 2.5 "the cluster resources, including parallel data services,..."), to combine

with Cramer reference's file mounting mechanism for providing teachings to reject specific limitations of the claims.

b). At Pages 20-21, the Applicants argued the “**determining ... whether further request was submitted ...**” concerning the limitations of claims 8 and 9 is noted and considered. By introducing SunC22 as the new primary reference and changing Cramer as the secondary, the Examiner has stated, in the supplemental non-final Office Action, the Cramer reference (at col. 5, lines 15-23) teaches pairing proxy vnodes on each client node with corresponding vnodes on the server side, the global unmount mechanism utilizes the vnode link to determine the node of the submitting request. The combined SunC22's teaching on parallel file system on cluster nodes and Cramer's on mounting request and response mechanism between computer nodes would have taught the first cluster server to determine the server which requests mounting or unmounting file system.

c). At Pages 22-23, the Applicants argued about the rejections of other dependent claims based on the argument for the independent claims 1, 18 and 35, the Examiner has noted and addressed the dependent claims as stated above.

### ***Conclusion***

#### 7. The prior art made of record

U. Sun Cluster 2.2 Software Installation Guide, Sun Microsystems, Inc. Part  
Number 806-1008, April 1, 1999

A. U.S. Patent No. 5,946,685

B. U.S. Patent No. 6,363,411

C. U.S. Patent No. 6,192,408

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

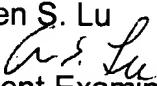
D. U.S. Patent No. 6,275,953

E. U.S. Patent No. 6,151,684

F. U.S. Patent No. 6,202,080

G. U.S. Patent No. 6,249,879

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuen S Lu whose telephone number is 703-305-4894. The examiner can normally be reached on 8 AM to 5 PM, Monday through Friday. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on 703-305-9790. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

Kuen S. Lu  
  
Patent Examiner

June 3, 2004

  
SRIRAMA CHANDRA MURTHY  
PRIMARY EXAMINER